
This is the **published version** of the article:

Benítez Bravo, Rubén; Liu, Xiao; Martín Mor, Adrià, dir. Proyecto de localización de Waterfox al catalán y al chino. 2019. 38 p.

This version is available at <https://ddd.uab.cat/record/203606>

under the terms of the  license

Trabajo de TPD

LOCALIZACIÓN DE WATERFOX EN>CA/ZH



Ilustración 1. Logo de Waterfox

Rubén Benítez Bravo y Xiao Liu

1332396 | 1532305

Ruben.benitez@e-campus.uab.cat

Xiao.Liu@e-campus.uab.cat

Màster de tradumàtica: Tecnologies de la traducció

2018-2019

Índice de contenidos

1. Introducción	4
2. Tabla resumen	5
3. Descripción del proceso de localización	6
3.1 ¿Qué es Waterfox?	6
3.2 ¿Qué uso tiene?	6
3.3 Preparando la traducción	7
4. Análisis de la traducción	11
5. Análisis técnico	21
6. Fase de testing	29
7. Conclusiones	35
8. Bibliografía	38

Índice de ilustraciones

Ilustración 1. Logo de Waterfox	1
Ilustración 2. Botón de descarga en GitHub	7
Ilustración 3. Alex Kontos dando el visto bueno	10
Ilustración 4. Contribuir al proyecto Waterfox	10
Ilustración 5. Alex Kontos dando su visto nuevo otra vez	11
Ilustración 6. Análisis de los archivos EN-ZH 1	11
Ilustración 7. Análisis de los archivos EN-ZH 2	12
Ilustración 8. Problemas con las mayúsculas y los espacios de más	13
Ilustración 9. Ejemplo de mayúsculas que no se deben repetir	13
Ilustración 10. Solución comillas latinas	14
Ilustración 11. Problema de codificación apóstrofo	14
Ilustración 12. Problema de codificación « · »	15
Ilustración 13. Ejemplo de inclusión de artículos y preposiciones	15
Ilustración 14. Detector de espacios y otros símbolos ocultos	15
Ilustración 15. Diccionario Glosbe	17
Ilustración 16. Linguee	17
Ilustración 17. Tutorial de OmegaT	18
Ilustración 18. OmegaT – Opciones	18
Ilustración 19. Tutorial OmegaT 2	19

Ilustración 20. OmegaT – Ir a buscar y reemplazar.....	20
Ilustración 21. OmegaT – Buscar y reemplazar	20
Ilustración 22. Ejemplo de error a evitar.....	22
Ilustración 23. Como elegir la concordancia en OmegaT	23
Ilustración 24. Herramientas en OmegaT	24
Ilustración 25. OmegaT – Revisión	24
Ilustración 26. Opciones de revisión en OmegaT.....	25
Ilustración 27. OmegaT – Control de calidad	26
Ilustración 28. OmegaT – Control de calidad 2	26
Ilustración 29. Configuración de codificación	27
Ilustración 30. Signos extraños	27
Ilustración 31. Signos extraños 2	28
Ilustración 32. Resultado de detección	28
Ilustración 33. Memoria en la carpeta correcta	28
Ilustración 34. Error 1 – Texto sin traducir	29
Ilustración 35. Error 2 – El mismo texto sin traducir	29
Ilustración 36. Error 3 – Más texto sin traducir	30
Ilustración 37. Error 4 – Problemas con las tildes	30
Ilustración 38. Error 5 – El mismo problema con las tildes	31
Ilustración 39. Error 6 – Problemas con las diéresis	32
Ilustración 40. Problema casi solucionado	32
Ilustración 41. Problema solucionado	33
Ilustración 42. Ejemplo de lo que también seríamos capaces de traducir (catalá).....	33
Ilustración 43. Ejemplo de lo que también seríamos capaces de traducir (chino).....	34

Índice de tablas

Tabla 1: Análisis sin la memoria de traducción	8
Tabla 2: Análisis después de aplicar la memoria de traducción	9

1. Introducción

En este trabajo hemos llevado a cabo un proceso de localización, tratando de ponernos en la situación de unos traductores profesionales que reciben un encargo en el que tienen que traducir un navegador web de inglés a catalán y a chino.

Entiéndase por «proceso de localización» la suma de todas las fases necesarias para traducir un producto digital, desde que se recibe el encargo hasta que se envía la traducción al cliente. Entre las fases incluiríamos la recepción del encargo, un análisis de lo que se va a traducir, la traducción en sí misma, la revisión y el envío del resultado final, entre otras.

Además, la localización conlleva que el producto se traduzca para un público en concreto. En nuestro caso hemos hecho una traducción al catalán para los usuarios de Cataluña y otra al chino para los usuarios de la República Popular China.

Al principio, Rubén iba a hacer el trabajo por sí solo. Empezó a planificar el proyecto en la universidad, cuando el profesor dejó un lapso de tiempo para empezarlo al final de una clase. Xiao se unió más tarde, cuando buscaba un compañero o grupo al que unirse para hacer el trabajo.

Rubén, aparte de traducir al catalán, se ha encargado de hacer de gestor del proyecto, de revisor y además de hacer el testing. Por otra parte, la tarea de Xiao ha consistido en traducir los archivos al chino, revisar sus traducciones, y también de hacer el testing.

Lo primero que se decidió fue el producto que se traduciría. Tenía que tratarse de un software libre, con código abierto, para de este modo poder modificar el código sin tener problemas de derechos o de acceso a los archivos traducibles.

Rubén decidió que una manera de decidirse por una herramienta era documentarse, buscar muchas opciones y finalmente escoger una. Encontró una extensa entrada de Wikipedia en inglés llamada [*List of free and open-source software packages*](#) que recoge centenares de ejemplos de herramientas con código abierto. De entre todas ellas, escogió Waterfox. Al unirse a él, Xiao estuvo de acuerdo en traducir ese programa. Sería un proyecto sencillo pero interesante.

No debemos olvidar que es un proyecto de localización. Se han utilizado herramientas TAO y otros recursos digitales como memorias o glosarios para llevar a cabo el proyecto. En las siguientes páginas describiremos con más detalles el proceso que se ha seguido.

2. Tabla resumen

Miembros: Rubén Benítez Bravo y Xiao Liu

Producto seleccionado: Waterfox Project - www.waterfoxproject.org

Combinación lingüística: EN>CA/ZH

Número de palabras (aproximadamente): 6.500

Breve descripción: Navegador web con código abierto basado en Mozilla disponible para sistemas operativos de Windows, Linux, MacOS y Android

Estrategia de traducción (TA/TAO) y programas requeridos:

Descargar los archivos a través del sitio web Github. Traducir los archivos mediante una herramienta TAO. Tener presente el navegador en inglés para evitar problemas de falta de contexto. Al acabar de revisar la traducción, hacer el testing. Y, lo más importante, aprender a localizar un producto digital

·[Memsources](#), [OmegaT](#)

¿Se ha establecido contacto con el equipo de desarrollo? Sí, y ha estado de acuerdo en que use su herramienta para este trabajo

3. Descripción del proceso de localización

3.1 ¿Qué es Waterfox?

Se trata de un navegador web basado en Mozilla desarrollado en el Reino Unido y con miles de usuarios en todo el mundo. Es muy similar a la versión ESR de Firefox, a la que se le sustituyó el panda rojo del logotipo por una especie de zorro acuático, hasta que en 2015 se estableció el logo actual: la silueta de [un zorro azul sobre una esfera celeste](#).

Indagando un poco más, descubrimos que este navegador alternativo a Firefox, que nació en el año 2011, se lanzó bajo una premisa: mejorar Firefox ESR y hacerlo más veloz. En aquel momento aquella versión de Firefox funcionaba a 32 bits, y la velocidad de Waterfox ya alcanzaba los 64 bits, aun habiendo salido el mismo año. No obstante, Mozilla ha ido sacando nuevas versiones de su navegador principal que suplen la lenta velocidad de esa versión anticuada, de modo que utilizar Waterfox de ahí en adelante dejó de tener mucho sentido para sus usuarios habituales. Utilizarlo equivaldría a tener una versión anticuada y un poco lenta de Firefox con algunos cambios menores, como tener deshabilitado por defecto el servicio Pocket para guardar páginas que el usuario desea abrir más adelante. Y también ciertas restricciones de seguridad y privacidad que impedirían, por ejemplo, utilizar Netflix. Con el tiempo fueron apareciendo versiones mejoradas del navegador de la mascota azul y ya no existen estas carencias, pero la gran desventaja que sigue arrastrando Waterfox es que las actualizaciones de seguridad tardan unos días más en llegar que las del navegador original de Mozilla.

Sin embargo, Waterfox aún está vivo y se siguen creando nuevas versiones. La más reciente, Waterfox 56.0.4, se lanzó en febrero de 2018, nueve días más tarde que las últimas versiones de Firefox (Firefox 58 y Firefox ESR 52.6), con los parches actualizados.

De modo que la única diferencia notable entre Firefox y Waterfox hoy en día, aparte de su popularidad, es la tardanza de las actualizaciones de este último en llegar, unos ajustes de seguridad predeterminados y que la herramienta resulta un poco más lenta, pero esta última disparidad es un cambio tan ínfimo que el usuario medio no notaría la diferencia.

3.2 ¿Qué uso tiene?

Con estos datos en mano, no es difícil imaginar que la popularidad de este navegador sea limitada, comparada con la de otros como Firefox, Chrome o Safari. Sin embargo,

que Waterfox sea una herramienta de código abierto poco conocida es una ventaja si lo que queremos es traducirla para practicar. Y si bien existen una versión en catalán y otra en chino disponibles, estas no están completamente traducidas. Así que, para dicho propósito, este navegador nos es perfecto.

Desde la entrada de Waterfox en [Wikipedia](#) encontramos un enlace a [Github](#), creado por Alex Kontos, el director del proyecto Waterfox: un joven británico de origen griego, un desarrollador de software que se encuentra cursando un máster en Ingeniería de Software en la Universidad de Oxford el presente curso 2018-2019. Con los archivos descargados desde ese enlace y descomprimidos en nuestro ordenador, podemos empezar a echarle un vistazo a las entrañas de la herramienta. También tenemos acceso al enlace desde la página web de [Waterfox](#).

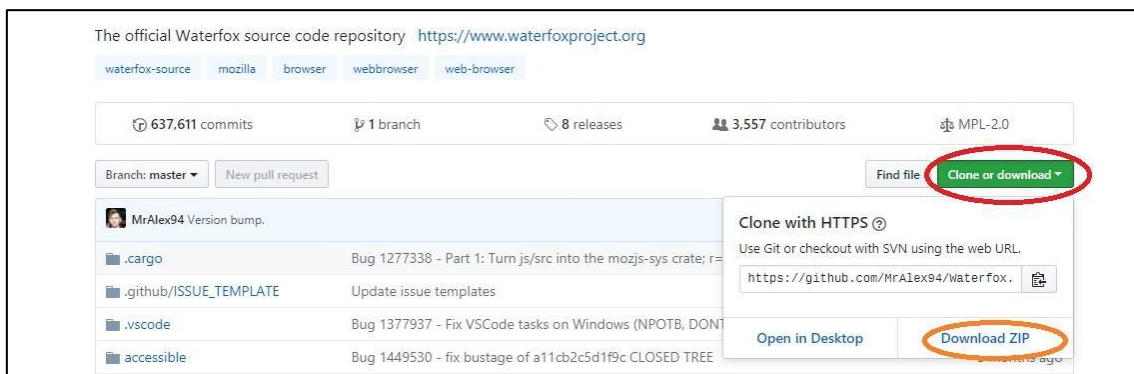


Ilustración 2. Botón de descarga en GitHub

3.3 Preparando la traducción

(Rubén)

Decidí utilizar Memsorce porque ya conocía esta herramienta de antes. A lo largo de la carrera recuerdo haberla utilizado en más de una ocasión en clases de informática. Es más, en una asignatura de cuarto de carrera vino una persona que trabajaba como poseditora y nos ofreció un ejercicio voluntario en el que tendríamos que poseditar segmentos traducidos por una herramienta de traducción automática. La recompensa era una licencia limitada de Memsorce que permitía abrir varios proyectos sin tener que pagar, ya que la licencia gratuita por defecto del programa limita al usuario a poder abrir solamente dos proyectos a la vez.

Si el usuario desea iniciar más de dos proyectos al mismo tiempo necesita conseguir una licencia superior, las cuales son de pago. Antes de acabar este trabajo no veía problema en seguir utilizando esta herramienta con esta licencia. Después de todo, es un software online, fácil de utilizar, visualmente atractivo y funcionalmente intuitivo.

Antes de nada, tuve que decidir qué parte del texto de la interfaz traduciría. Para ello me descargué el programa en inglés y estuve observando el texto en pantalla (el mensaje de bienvenida, la barra de herramientas, las opciones, etc.). Tras buscar durante un buen rato el texto que veía en pantalla en GitHub, di con dos extensiones de archivos que contenían esa información, «.properties» y «.dtd», así que descargué los 60 archivos que había con esas extensiones.

Lo primero que hice al abrir Memsources fue crear un proyecto, que bauticé como «Waterfox». Era un proyecto con el inglés como idioma fuente y el catalán como idioma meta. No puse ninguna fecha de entrega e importé los archivos recién descargados.

A continuación hice un análisis completo, sin pretraducción y sin añadir ningún recurso extra. Esta tabla ilustra el resultado del primer análisis de los archivos:

	Segments	Pages	Words	Characters	%
Net Rate	1507	21.49	6541	33641	
All	2168	23.76	7523	37414	100
Repetitions	631	2.06	917	3416	12.2
101%	0	0	0	0	0
100%	31	0.05	28	88	0.4
95%–99%	98	0.44	165	724	2.2
85%–94%	19	0.29	83	461	1.1
75%–84%	58	1.05	322	1625	4.3
50%–74%	353	4.64	1437	7276	19.1
0%–49%	978	15.23	4571	23824	60.8

Tabla 1: Análisis sin la memoria de traducción

El siguiente paso era crear una memoria de traducción. Hasta ahora no había creado ninguna con esa cuenta nueva. De modo que creé una con combinación inglés-catalán, la llamé «Memòria EN-CA» y la preparé para el proyecto. Decidí investigar si existía alguna memoria en softcatalà que pudiera serme útil y encontré una de Mozilla que me vendría muy bien. La descargué y la importé a la memoria que acababa de crear.

Después volví a hacer un análisis, esta vez aplicando la memoria recién creada. Por supuesto, quise que la máquina hiciera una pretraducción y poder descargar, una vez terminado el proyecto, la memoria actualizada junto con los archivos bilingües y las traducciones. Escogí la opción de reconocer las repeticiones y segmentos intraducibles, hacer un control de calidad que incluyese detalles como la puntuación agregada, que corrigiese las mayúsculas y otras optimizaciones, como que pretradujese los segmentos incluidos en la memoria a partir de un 75 % de concordancia. Y este fue el resultado:

	Segments	Pages	Words	Characters	%
All	2168	23.76	7523	37414	100
Repetitions	0	0	0	0	0
101%	606	9.83	3069	15237	40.8
100%	1553	13.83	4412	22023	58.6
95%–99%	2	0.01	3	13	0
85%–94%	0	0	0	0	0
75%–84%	1	0.03	14	48	0.2
50%–74%	0	0	0	0	0
0%–49%	6	0.06	25	93	0.3

Tabla 2: Análisis después de aplicar la memoria de traducción



Ilustración 3. Alex Kontos dando el visto bueno

(Xiao)

Al principio, para ver el avance actual de la traducción de los programas, consulté dos plataformas: Pontoon y Transifex.

También podemos entrar en la página oficial de Waterfox y hacer clic en «Contribute».

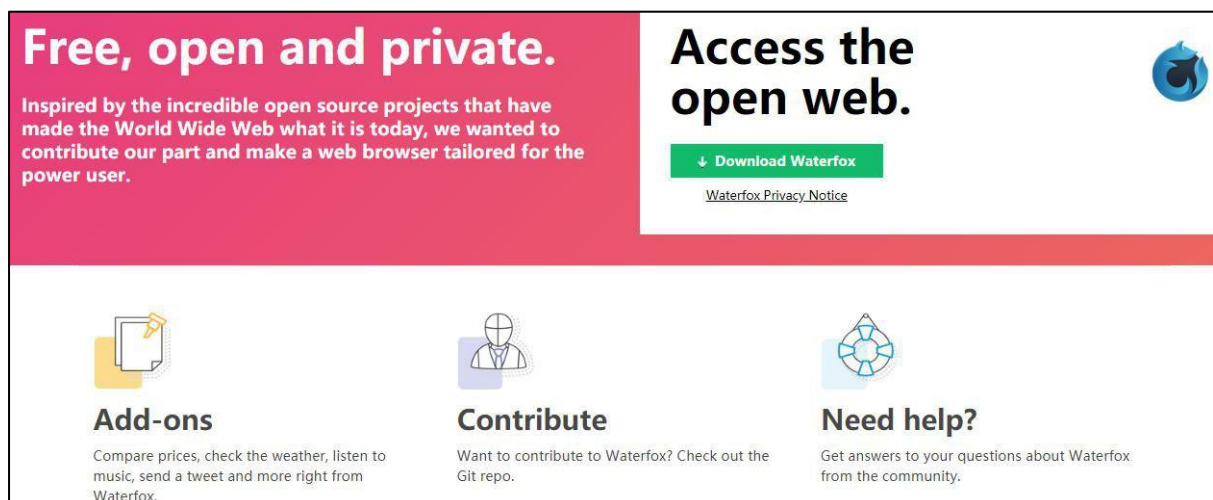


Ilustración 4. Contribuir al proyecto Waterfox

Para empezar mi trabajo, primero fui a Github a descargar Waterfox en versión máster desde la página web: <https://github.com/MrAlex94/Waterfox>. Para traducirlo al chino he utilizado la herramienta TAO: OmegaT.



Ilustración 5. Alex Kontos dando su visto nuevo otra vez

Como los primeros pasos en cualquier herramienta de traducción, creé un nuevo proyecto «Project Waterfox En-Zh». Después de haber consultado a Rubén las razones por las que eligió los archivos con extensiones «.dtd» y «.properties», los importé en mi proyecto de OmegaT.

Lo primero que hice fue un análisis básico del documento:

Estadísticas del proyecto					
	Segmentos	Palabras	Caracteres (sin espacios)	Caracteres (con espacios)	#Archivos
Total:	2.157	7.712	38.226	43.638	60
Pendientes:	25	69	326	371	16
Únicos:	1.529	6.765	34.756	39.869	60
Pendientes Únicos:	21	64	306	349	14

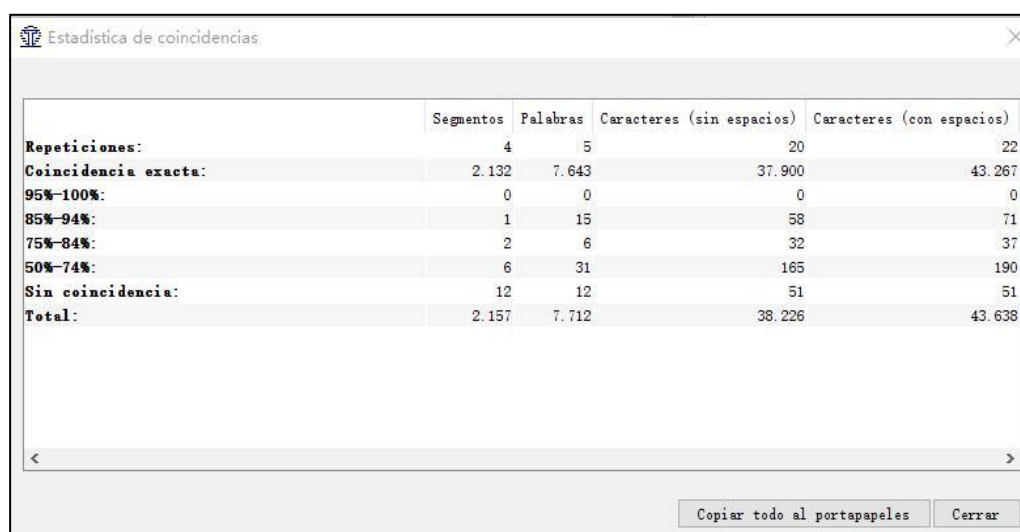
Estadísticas por archivo individual:				
Nombre de archivo	Total de segmentos	Segmentos pendientes	Segmentos Únicos	Segm
要翻的内容2/aboutPrivateBrowsing.properties	2	0	2	^
要翻的内容2/accounts.properties	26	0	25	
要翻的内容2/advanced.dtd	98	0	84	
要翻的内容2/applicationManager.dtd	4	0	3	
要翻的内容2/applicationManager.properties	6	0	6	v

Ilustración 6. Análisis de los archivos EN-ZH 1

Considerando que Waterfox es muy similar a Firefox, para facilitar la traducción, descargué el TMX inglés-chino de Firefox desde la plataforma Pontoon, la página web es

<https://pontoon.mozilla.org/zh-CN/firefox/team-info>. Y luego lo importé a mi proyecto de OmegaT.

Luego hice un análisis de concordancias con la memoria que había creado:



	Segmentos	Palabras	Caracteres (sin espacios)	Caracteres (con espacios)
Repeticiones:	4	5	20	22
Coincidencia exacta:	2.132	7.643	37.900	43.267
95%-100%:	0	0	0	0
85%-94%:	1	15	58	71
75%-84%:	2	6	32	37
50%-74%:	6	31	165	190
Sin coincidencia:	12	12	51	51
Total:	2.157	7.712	38.226	43.638

Ilustración 7. Análisis de los archivos EN-ZH 2

Más tarde, empecé a configurar los motores de traducción automática, pero al final no los he usado por las cuestiones de pago y la limitación del idioma. Luego hice las traducciones de los 60 archivos, corrigiendo las concordancias menos exactas. Después de haber terminado al 100 % las traducciones, hice un control de calidad y exporté los archivos traducidos, las memorias de traducción y el glosario generado. Luego, hice el testing en la ventana del *dialog* « Waterfox». Al final, se lo entregué todo mi trabajo a Rubén.

4. Análisis de la traducción

(Rubén)

En general, el hecho de utilizar una memoria tan extensa hizo que, dentro de los segmentos traducidos y bloqueados automáticamente, una revisión no fuese necesaria más que en casos concretos, como los de concordancia de género y número. Ahora bien, hubo segmentos bloqueados por Memsource que contenían errores. Por ejemplo, en el original había una letra mayúscula y en el meta una minúscula, y hubo que corregirlos mediante el previo desbloqueo con un ctrl+l. En otros casos había un espacio de más que la máquina introdujo automáticamente. Debieron de ser cosas de la memoria, ya que no utilicé traducción automática en este proyecto.

72	A	a	✓	100
73	Remove·All·Shown	·Suprimeix·totes·les·mostrades	✓	100

Ilustración 8. Problemas con las mayúsculas y los espacios de más

A decir verdad, lo que estaba haciendo durante buena parte de la traducción no era traducir sino corregir segmentos que la máquina traducía por mí y que me han ahorrado muchísimo tiempo. Por ejemplo, con las etiquetas o con texto que se va repitiendo, el cual la herramienta va traduciendo automáticamente al detectar que hay una repetición. Aparte, tras corregir un error, como podría ser la inclusión de un artículo antes de un sustantivo, la máquina lo irá incluyendo en los segmentos similares que vayan apareciendo.

Y otra característica positiva es que las traducciones generadas por la memoria no incluyen las mayúsculas propias del inglés para destacar ciertos nombres comunes. O al menos no lo hace la mayoría de las veces.

22	Control·Sharing	Control·de·compartició	✓	100
----	-----------------	------------------------	---	-----

Ilustración 9. Ejemplo de mayúsculas que no se deben repetir

Para otra ocasión recurriré a crear un glosario donde almacenar estos elementos que no debemos traducir y de ese modo evitar traducirlos inconscientemente y, más grave todavía, generar traducciones diferentes. Por despiste o por ir con prisas cualquier traductor puede cometer este error de principiante.

Un detalle que percibí y que agradezco es que esta herramienta acepta caracteres propios de otras lenguas que no sean la inglesa, como las comillas latinas. Pero no los acepta todos por cuestiones de codificación, como el apóstrofo, el cual tuve que corregir en la opción «formato» en la barra de herramientas.

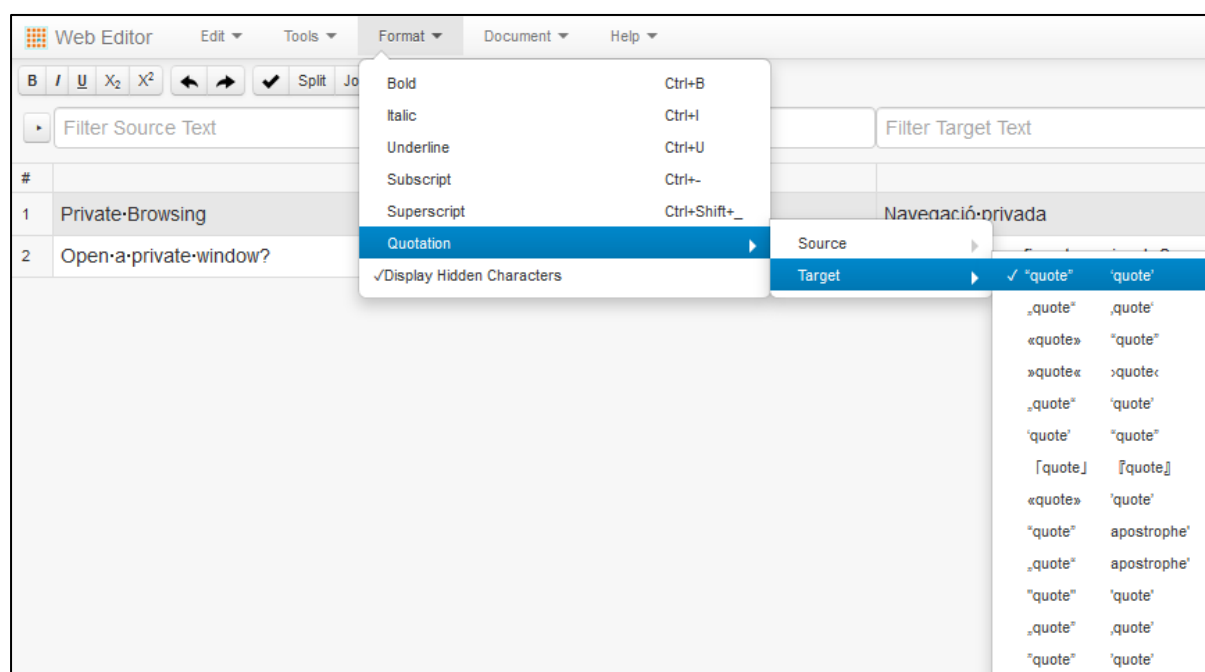


Ilustración 10. Solución comillas latinas

Como aspecto a tener en cuenta, modificar el formato trajo la consecuencia de que los apóstrofes de los segmentos originales aparecieran así «â», y hubo que ir con cuidado de que no se escapara ninguna «â» en las traducciones. Y de hecho la memoria no traducía todos los apóstrofes, lo que se arreglaría en la fase de control de calidad.

9	%S-prevented-this-site-from-asking-you-to-install-software-on-your-computer.	El-%S-ha-evitat-que-aquest-lloc-us-demani- instal-lar-programari-al-vostre-ordinador.	✓	100	🔒
10	Donât-Allow	No-permeteu	✓		⬇️

Ilustración 11. Problema de codificación apóstrofo

Pero hubo un símbolo que Memsources no logró identificar correctamente: el punto de la «l·l», el cual consideraba que es un «.». Incluso después de corregir el formato, la «l·l» seguía apareciendo como un error.

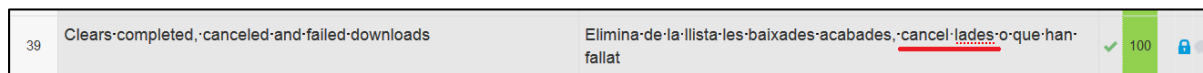


Ilustración 12. Problema de codificación «l·l»

Además, los artículos y los numerales indefinidos no aparecen en las traducciones generadas por la memoria porque el inglés no los necesita, pero el catalán sí. Hay que añadirlos, y cambiar de sitio las etiquetas si es necesario para que la frase tenga sentido.

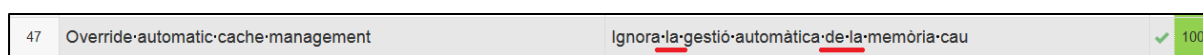


Ilustración 13. Ejemplo de inclusión de artículos y preposiciones

Gracias al comando ctrl+insert, se pueden copiar las etiquetas o los enlaces de forma casi automática. Aparte, me di cuenta de que es importante aprender a leer las etiquetas, para descubrir qué es texto y qué es comando, ya que Memsources no siempre los sabe separar. Por fortuna, en estos archivos la herramienta no mostraba el código.

También recomiendo activar el detector de espacios y otros símbolos ocultos para verlos mejor, y así evitar dejarnos algún espacio. En casi cualquier herramienta en la que se pueda editar un texto puede activarse y desactivarse haciendo clic en un botón. En la siguiente imagen se puede ver como es el botón en Memsources.

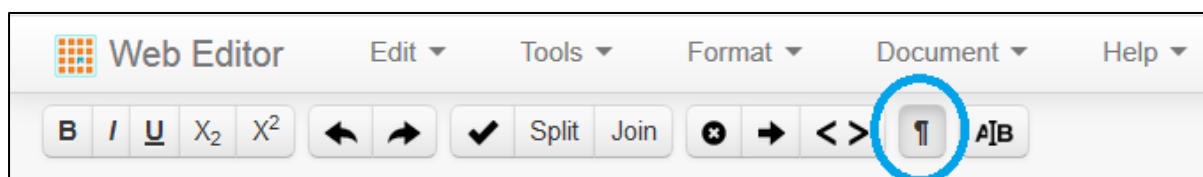


Ilustración 14. Detector de espacios y otros símbolos ocultos

Gracias a otros atajos, como el ctrl+(un número), pude añadir las sugerencias de la memoria y también de las bases terminológicas, glosarios o traductores automáticos, si los estuviera utilizando. Además, con el ctrl+ se añaden las etiquetas en un momento. De este modo el usuario no se olvida de incluir ninguna.

En lo que respecta a resolver dudas lingüísticas utilicé el recurso online [Optimot](#) y el diccionario [Diec](#).

Decidí usar el tratamiento informal (*vosaltres*) y la segunda forma del plural del imperativo (*tingueu*). Así aparecía en la memoria y es habitual verlo en software traducidos. No obstante, a veces el imperativo se traducía en primera persona del singular y otras en segunda del plural. Hubo que unificarlo. Para estos problemas seguí la guía de estilo de [softcatalà](#), mismo portal de donde saqué la memoria (véase «Memòria EN-CA»). Por lo tanto, una de las cuestiones que he tenido en cuenta ha sido tratar de neutralizar el género de las palabras que hacían referencia a personas. Por ejemplo, si el programa se dirige al usuario.

Otra decisión que tomé fue escribir «internet» en minúscula. Sé que en catalán se aceptan ambas formas, en mayúscula y en minúscula, pero teniendo en cuenta que es un nombre común de un medio, como lo son la prensa, la radio o la televisión, no entiendo por qué hay quien lo sigue escribiendo en mayúscula en catalán, castellano y otras lenguas romances. Mediante una búsqueda en cada archivo, desde el buscador interno de Memsources, pude encontrar esa palabra y cerciorarme de que no estaba en mayúscula.

Una vez traduje los segmentos, tocaba hacer una revisión, antes de exportar los archivos del proyecto y la memoria. La primera revisión fue sobre los mismos fragmentos en la herramienta Memsources. Detecté bastantes fallos debidos a despistes. Pero para asegurarme de que el resultado fuera impecable, tenía que hacer algo más.

Cuando acabé, pulsé F7 para pasarle el corrector ortográfico a cada archivo. No me marcó muchos errores, solo repeticiones incorrectas debidas a aceptar las sugerencias de la memoria que solo eran concordancias parciales.

Por último pulsé F9 para aplicar el control de calidad antes de exportar los archivos finales. Vi que se me había olvidado incluir algunas comillas latinas y etiquetas. Estableceré unos parámetros diferentes para la pretraducción la próxima vez para que esto no vuelva a ocurrir, en la medida de lo posible.

(Xiao)

En mi traducción utilizo el tratamiento de usted porque en chino, cuando nos comunicamos con el cliente, siempre le llamamos de «usted», que se escribe así: «您».

Y en chino, solo se diferencian el género masculino y el femenino entre los objetos vivos, en palabras como «aplicación» o «navegador» no se distingue el género. Por eso, cuando traduje el Waterfox, no he diferenciado el género.

Además, en chino a veces se diferencian el singular y el plural y a veces no, dependiendo de la situación. Principalmente hay dos casos. En el primer caso, si el sustantivo se coloca después del numeral, la «s» que representa el plural no se traduce. Por ejemplo, en la traducción de Waterfox, «two» en chino es «二», «tag» en chino es «标签», «two tags» se traduce como «二 标签». Es decir, en este caso «tag» o «tags» se traduce siempre como «标签», tanto si en el original es en singular o en plural. En el segundo caso, si el sustantivo se coloca después del adjetivo demostrativo, la «s» necesita estar en la traducción. Por ejemplo, «esta configuración» se traduce como «这个配置», mientras que «estas configuraciones» se traduce como «这些配置». En chino, la traducción « » se ha convertido en « », que equivaldría a la «s» del inglés.

Por otra parte, si mientras traducía no había ninguna concordancia en la memoria de Firefox para algún segmento, consultaba el diccionario [Glosbe](#) o [Linguee](#).



Ilustración 15. Diccionario Glosbe



Ilustración 16. Linguee

Además, OmegaT nos permite a integrar TA, no la he usado por cuestiones de pago y la limitación del idioma, pero aquí explico cómo usarlas para que más personas lo sepan.

Para configurarla hay que seguir las instrucciones del siguiente enlace: http://omegat.org/howtos/google_translate.

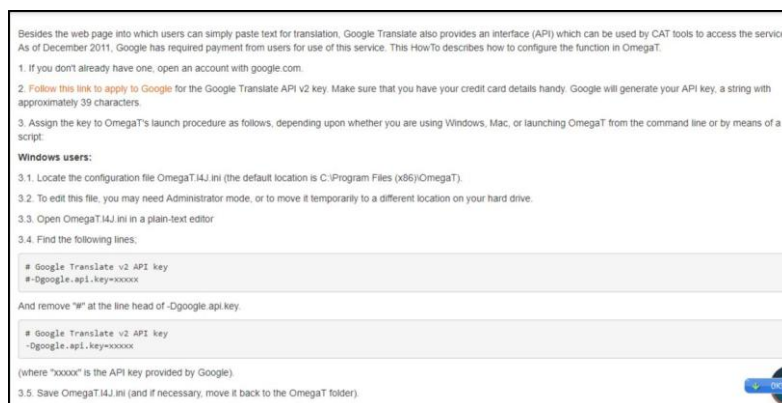


Ilustración 17. Tutorial de OmegaT

A continuación volvemos a OmegaT. Al hacer clic en la pestaña «Opciones» ó «Traducción Automática» ó «Google Translator v2» se puede usar Google Translator desde el programa. Pero la traducción automática de Google es un servicio de pago (costaría 20 dólares cada millón de caracteres).

También se puede introducir Microsoft Translator, siguiendo unas instrucciones similares. La ventaja de este otro servicio es que es gratuito, pero tiene un límite de 2 millones de caracteres por mes.

La misma situación se repite si se quiere usar Apertium, pero por desgracia solo traduce entre estos idiomas: CA, ES, GA, PT, OC y FR. El chino no está incluido, de modo que no me serviría.

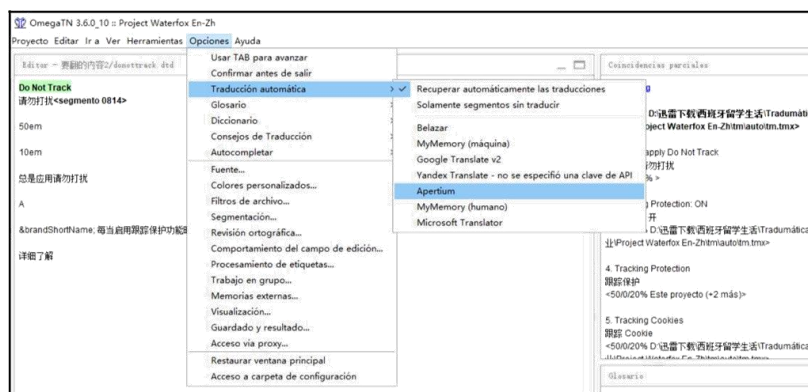


Ilustración 18. OmegaT Opciones

En OmegaT también podemos introducir diccionarios, como por ejemplo Merriam Webster 10th dictionary. Desde la siguiente dirección tenemos acceso a las instrucciones sobre cómo utilizar un diccionario desde esta herramienta TAO: <https://omegat.sourceforge.io/manual-latest/es/chapter.dictionaries.html>.

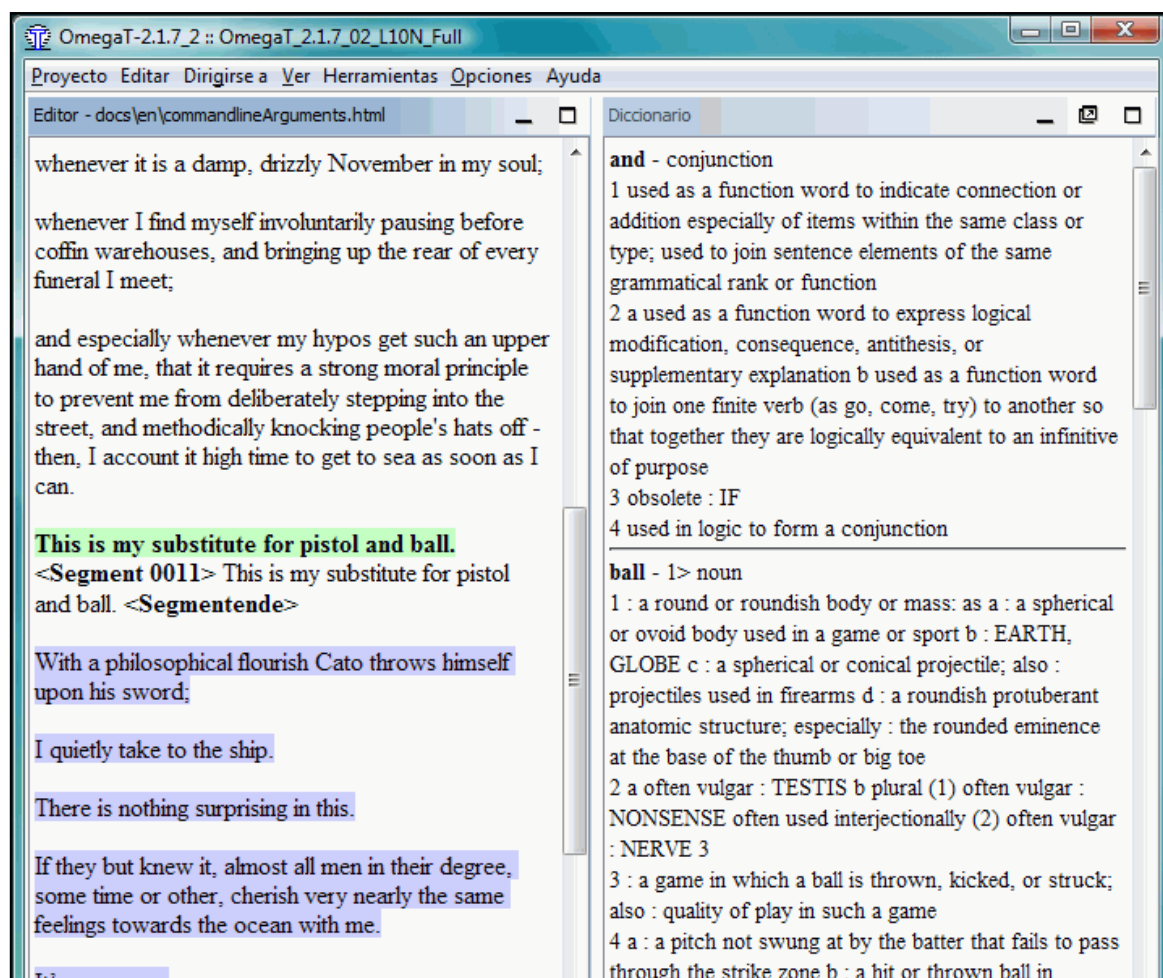


Ilustración 19. Tutorial OmegaT 2

En cuanto a corregir las traducciones incorrectas repetidas, podemos clicar en «Edit» ó «Search and Replace», para sustituirlas de una sola vez.

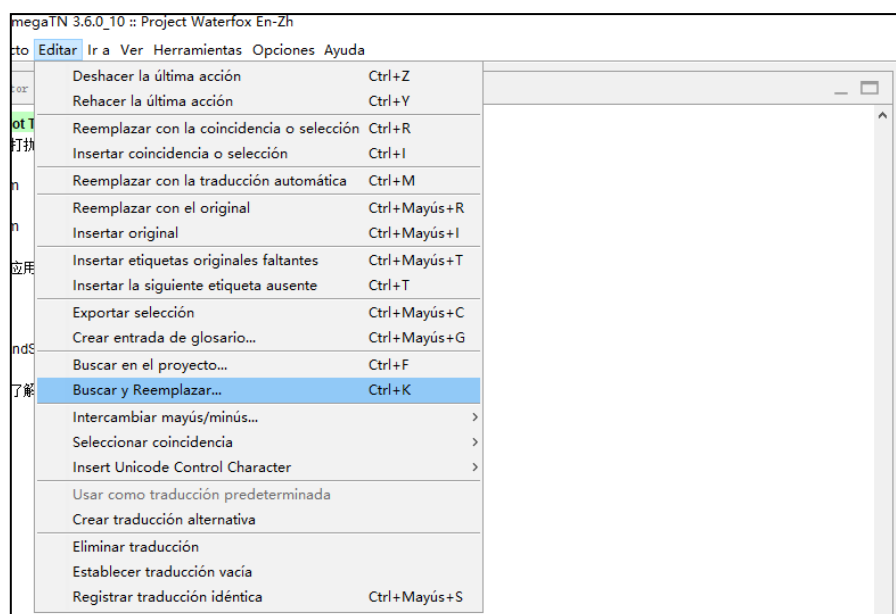


Ilustración 20. OmegaT \bar{E} Ir a buscar y reemplazar

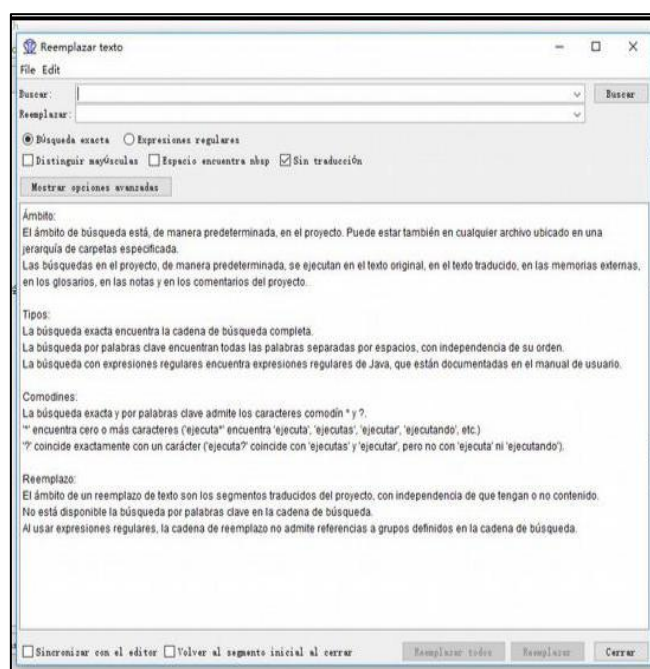


Ilustración 21. OmegaT \bar{E} Buscar y reemplazar

Por último, después de haber traducido el 100 % todos los segmentos, se generaron mi glosario completo y mi memoria, los cuales no solo sirven como referencia para la traducción posterior de este proyecto, sino también para la futura traducción de otros navegadores.

5. Análisis técnico

(Rubén)

Una ventaja que tiene Memsorce es que es una de esas herramientas TAO que separa el código y el texto traducible de una manera bastante eficaz. Esto evita que el usuario tenga que buscar otra herramienta que haga de filtro, en la mayoría de los casos. En este trabajo no he tenido problemas, pero Memsorce no siempre identifica del todo bien qué es código y qué es texto, y en caso de que aparezcan segmentos con código conviene utilizar el comando ctrl+insert para que más tarde el producto funcione con normalidad. Otra opción para solucionar el problema del código no detectado podría ser preparar los archivos antes de traducirlos. O también aplicar filtros diferentes, lo cual puede hacerse con otras herramientas para traducir software como Passolo. En cualquier caso, no ha sido necesario en los archivos que hemos traducido, ya que casi todo el contenido está compuesto por texto.

Tampoco me ha hecho falta convertir los archivos a ningún formato para que se cargasen correctamente. Por ese motivo opino que Memsorce es una herramienta muy cómoda para traducir sobre texto.

Otra ventaja es que se pueden extraer los archivos traducidos y las memorias de traducción en formato estándar, de modo que no es necesario convertirlos previamente. Asimismo, se pueden descargar los archivos bilingües o hacer un análisis directamente desde la página web y que aparezca en forma de tabla, por lo que tampoco es necesario buscar otra herramienta para ello.

Además, al ser una herramienta online, el progreso se guarda automáticamente cada pocos segundos. Esto implica que sea difícil que se pierda el trabajo ya hecho en caso de que, por ejemplo, se produzca un apagón. Y se puede trabajar desde cualquier ordenador sin necesidad de descargar ningún programa ni tener los archivos en el disco duro una vez se hayan subido al proyecto.

Ahora bien, el hecho de que sea online también tiene desventajas: si la conexión a internet da problemas, el programa irá lento y eso le quitará tiempo al traductor. Y además está el tema de la privacidad. Debido a que herramientas como Memsorce guardan los archivos cargados en la nube, existe la posibilidad de que se substraigan. Si estuviéramos trabajando en un proyecto confidencial podríamos tener problemas en ese aspecto.

Otro detalle es que se nos pedía que entregásemos la memoria que habíamos utilizado actualizada, en mi caso la ya mencionada «Memòria EN-CA», pero solo los fragmentos que habíamos modificado. La memoria que descargué al acabar el proyecto contenía tanto mis fragmentos como los que ya estaban en la memoria de Mozilla colgada en softcatalà. Para editarla utilicé el programa Notepad++, ingeniándomelas como podía para encontrar los fragmentos buscando mi nombre de usuario y borrando los que no lo contenían. No obstante, a modo de apunte para el futuro, se puede automatizar el proceso con herramientas como Olifant.

Uno de los problemas técnicos que más me costó solucionar fue el tema de las letras asignadas a los atajos para acceder desde el teclado, por ejemplo, a un botón de la barra de herramientas. Las letras estaban cambiadas en la memoria, y muchas veces se repetían. Gracias al control de calidad, que detecta este tipo de repeticiones, logré escoger letras que no estuvieran repetidas en cada archivo y que las letras escogidas estuvieran presentes en la palabra o palabras del botón. Claro que si una letra estaba repetida en dos extremos del archivo sería difícil que crease problemas al ejecutar el programa. Más tarde, al hacer la

6. Fase de *testing*», de la cual hablaremos más adelante, no hubo ningún atajo repetido dentro del mismo cuadro de botones.

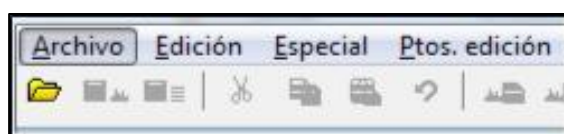


Ilustración 22. Ejemplo de error a evitar

(Xiao)

Elegí OmegaT para hacer la traducción, por las siguientes razones:

En primer lugar, para poder traducir los archivos con extensiones de .dtd y .properties. OmegaT es uno de los pocos que las reconoce y con el que puedo trabajar en chino. Para chequear los formatos compatibles con esta herramienta se puede consultar [este enlace](#).

En segundo lugar, en OmegaT no se necesita exportar manualmente la TMX actualizada. La herramienta genera automáticamente la TMX en una subcarpeta dentro de la carpeta del proyecto.

En tercer lugar, OmegaT guarda automáticamente el progreso, incluso cuando se están editando las traducciones, así que no es necesario guardarlas manualmente. Ni existe el riesgo de que se pierda la traducción si el ordenador se apaga de repente.

En cuarto lugar, la interfaz es muy sencilla y fácil de entender. En el lado derecho aparecen las opciones de coincidencias y se puede seleccionar cualquiera con solo un clic o con *shortcut keys*.

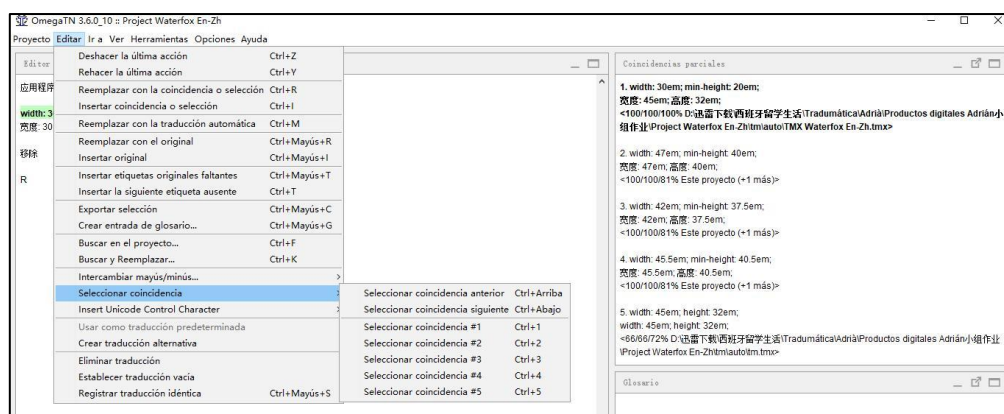


Ilustración 23. Como elegir la concordancia en OmegaT

En quinto lugar, hay que tener en cuenta que OmegaT, después de haber terminado la traducción, genera hasta tres TMX:

La primera, (level1), solo incluye las concordancias, y se puede usar en otra herramienta de traducción.

La segunda, (level2), incluye las concordancias y la etiquetas, y también se puede usar en otra herramienta de traducción.

Por último tenemos una memoria que solo incluye las concordancias y solo se puede reutilizar en OmegaT. Si se mueve a la carpeta «tm» podría usarse de nuevo en otros proyectos siempre que sea con esta misma herramienta TAO.

Nota: esta función varía según las diferentes versiones de OmegaT.

Y en sexto lugar, es gratuito y de código abierto.

En mi parte de trabajo, principalmente me encontré con tres problemas técnicos en lo que respecta a la traducción. También me encontré con problemas en la fase de testing junto con Rubén (véase

6. Fase de *testing*).

El primer problema consistió en el control de calidad. La versión de OmegaT que descargué fue la versión 3.6.0_10. Esa versión solo permite hacer validación de las etiquetas, pero no puede hacer la revisión ortográfica ni de terminología. Intenté solucionar este problema de muchas formas, al final encontré la solución.

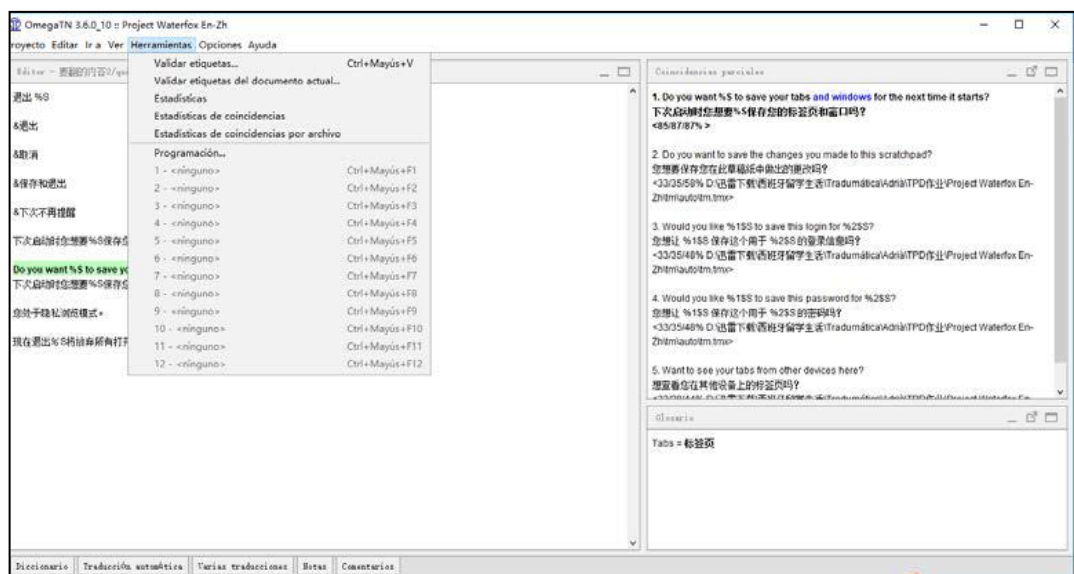


Ilustración 24. Herramientas en OmegaT

Resulta que la versión 4.1.5_3 contiene la función del control de calidad. Hice clic en «Herramientas» ó «Check Issues» y me salió una ventanilla que me hizo elegir qué tipo de revisión quise hacer. Seleccioné todos los campos.

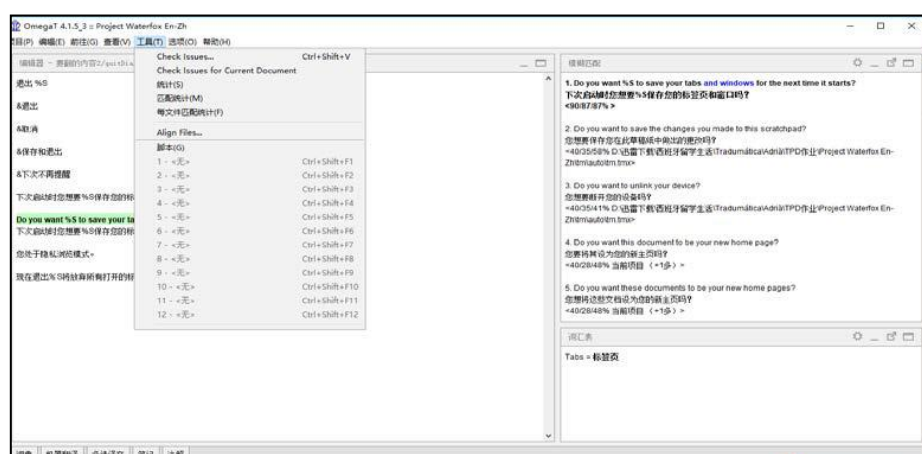


Ilustración 25. OmegaT E Revisión

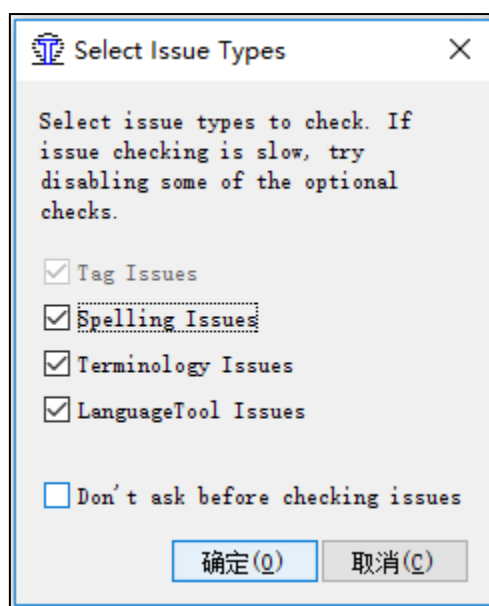


Ilustración 26. Opciones de revisión en OmegaT

Después se abrió una ventana indicándome que hubo 91 errores. Corregí los errores siguiendo las instrucciones. Sin embargo, algunas instrucciones no me parecieron razonables, en esos casos no modifiqué mis traducciones. Por ejemplo, mi traducción «过时的插件» correspondiente a «Outdated plugin» es correcta, pero me sugería erróneamente «过时地插件». La diferencia entre «的» y «地» consiste en que «的» se usa entre el adjetivo y el sustantivo, mientras que «地» se usa entre el adverbio y el verbo. La palabra «插件» es un sustantivo, entonces se debe usar «的», pero aquí se dividió mal el segmento y solo reconoció un carácter «插» en vez de una construcción de caracteres «插件». «插» en chino es un verbo, por tal razón OmegaT me recomendó equivocadamente «地».

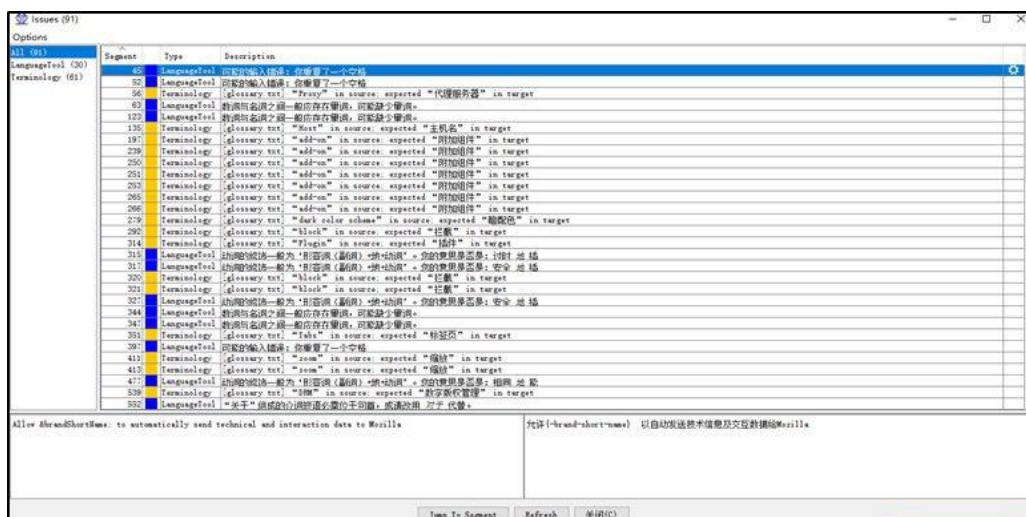


Ilustración 27. OmegaT É Control de calidad

Al final, después de haber corregido las que yo debía corregir y haber guardado las traducciones correctas como tal, solo quedaron 48 instrucciones, que no necesitan modificarse.

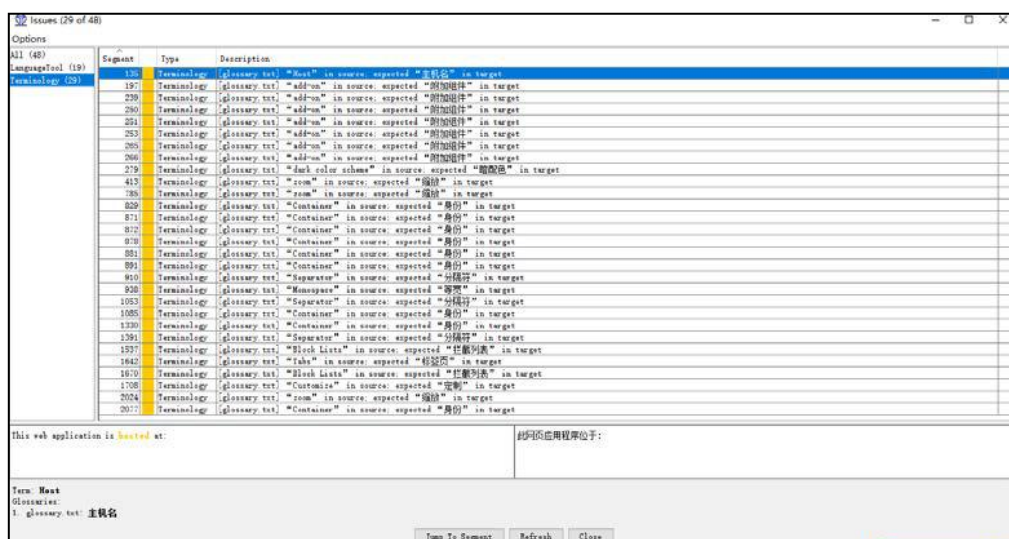


Ilustración 28. OmegaT É Control de calidad 2

Después de haber hecho el control de calidad, descubrí el segundo problema técnico. Aunque había traducido los archivos con extensión «.properties», no me salió la traducción cuando los abrí en Notepad++. Después de consultarlo con el profesor, solucioné este problema. La traducción no me salió porque en OmegaT no se ha configurado la codificación que se adapta al chino. Entonces, fui a «Opciones» - «Filtro de archivos» - «Paquete de

recursos Java(TM)» - «Codificación de archivo» - «UTF-16», luego hice clic en «Proyecto» - «Crear documentos traducidos». Con esta manera, conseguí abrir los archivos traducidos en chino con extensión «.properties» en Notepad++.

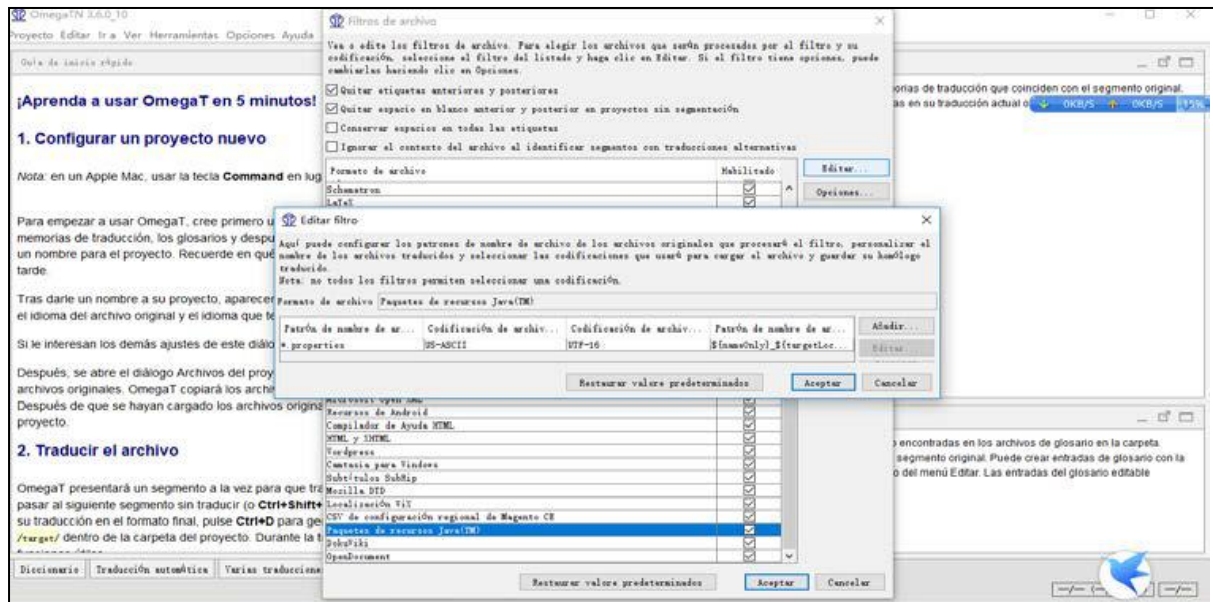


Ilustración 29. Configuración de codificación

El tercer problema ocurrió cuando me encontré con los signos que OmegaT no reconoció. Al principio los copié en mi traducción. Pero cuando entendí qué significaban los signos extraños, puse directamente los correspondientes signos en la traducción.

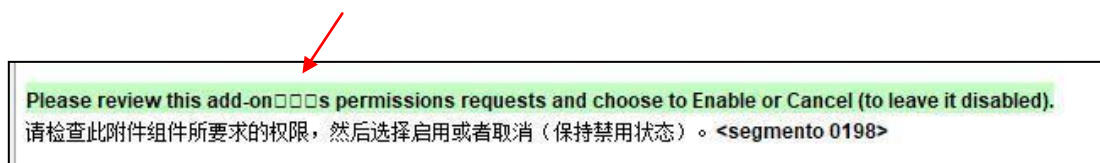


Ilustración 30. Signos extraños

En caso de encontrar símbolos que el sistema no reconoce, estos pueden copiarse y pegarse en el segmento meta, pero se puede intentar deducir de qué símbolos se trata. En estos casos es bastante intuitivo. En el primer caso se trataba de un apóstrofo y en el segundo de unas comillas.

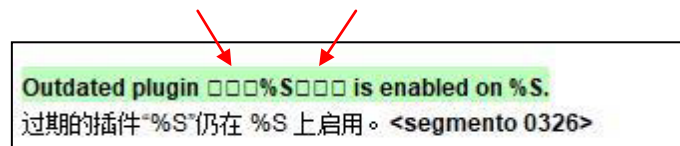


Ilustración 31. Signos extraños 2

Para comprobar que no hay ningún error en nuestro segmento se puede ejecutar el comando ctrl+shift+v. Esto hará que se abra automáticamente una ventanilla que detecta si hay algún error. No había errores en ninguno de los dos casos.

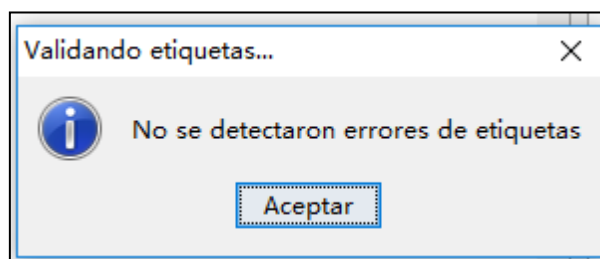


Ilustración 32. Resultado de detección

Advertencia: Cuando añadí la memoria de traducción de Firefox a la carpeta «tm», surgió un problema. La concordancia salió manualmente, pero no automáticamente, es decir, solo salió la concordancia del segmento que estoy traduciendo, pero no las concordancias de todos los segmentos automáticamente. Este inconveniente se debió a que puse la memoria de Firefox en la carpeta de «tm», pero no la puse en la carpeta «auto», la subcarpeta dentro de «tm». Aquí pongo este ejemplo para que otras personas no vuelvan a cometer el mismo error.

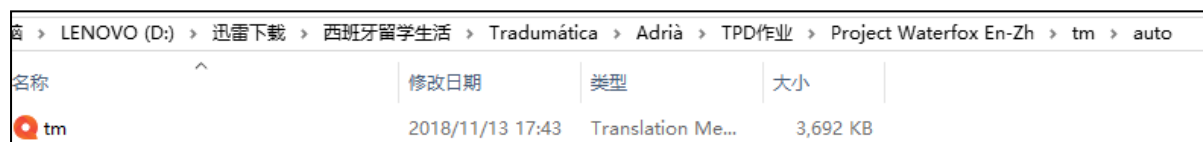
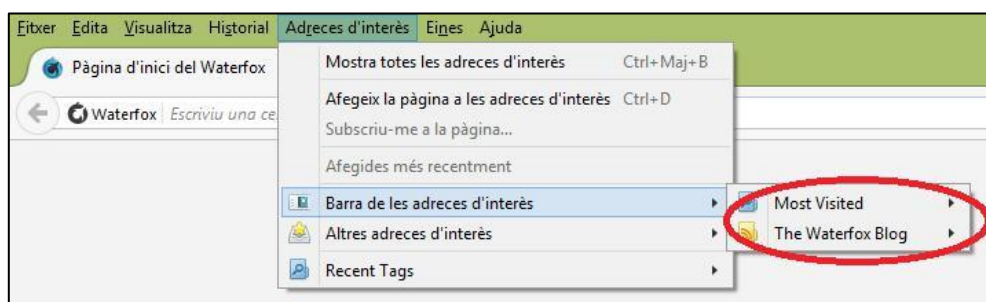


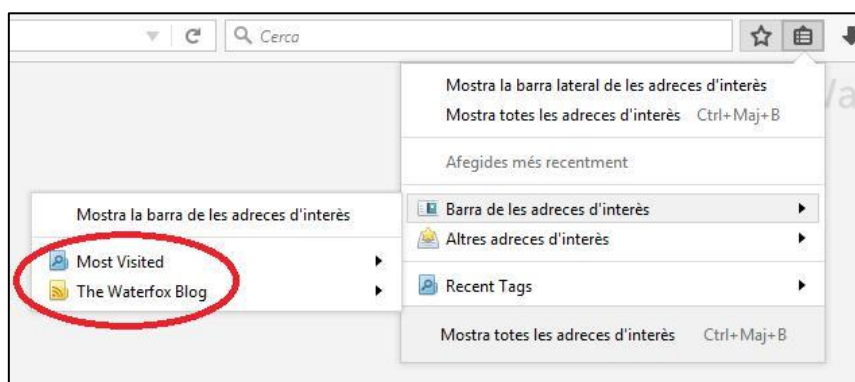
Ilustración 33. Memoria en la carpeta correcta

6. Fase de testing

Enlazado con la cuestión de los errores, en la versión ya existente en catalán de Waterfox existen algunos que pueden verse nada más abrir el navegador y empezar a explorarlo. A continuación se muestran algunos ejemplos en las siguientes capturas.



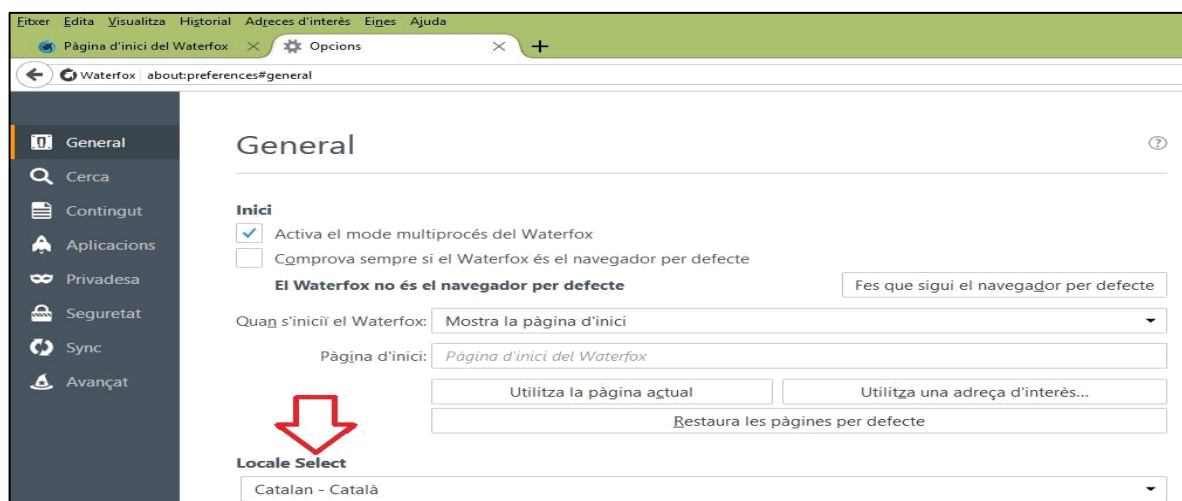
Il·lustració 34. Error 1 È Texto sin traducir



Il·lustració 35. Error 2 È El mismo texto sin traducir

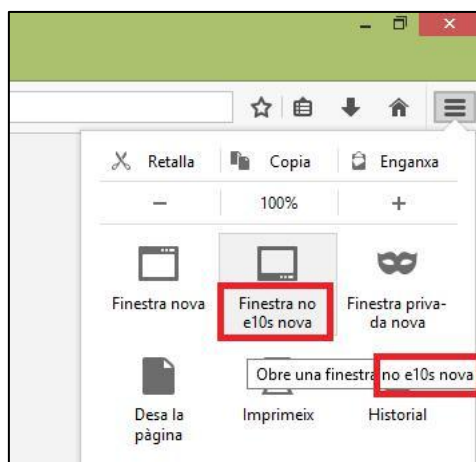
Como podemos ver, hay texto sin traducir en el navegador. En este caso, el apartado "Marcadores" en castellano y "Bookmarks" en inglés) puede que no estuviera en los archivos que enviaron a los traductores de esta versión.

Para cambiar el idioma de la interfaz hay que ir al apartado opciones, el cual también contiene elementos no traducidos.



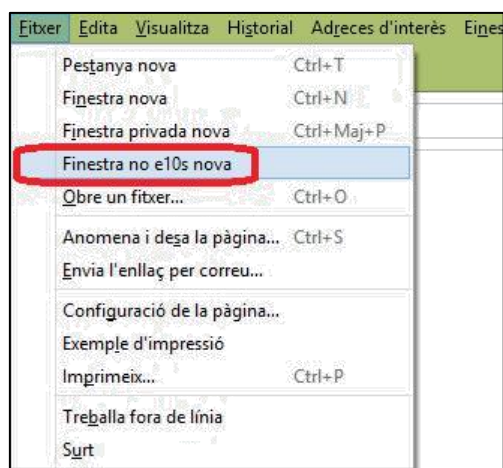
Il·lustració 36. Error 3 E Más texto sin traducir

Y, aparte, encontré estos otros dos. Como se puede ver en las siguientes ilustraciones, los errores no solo consisten en texto sin traducir, sino en problemas con la aceptación de caracteres, como las palabras acentuadas, debido seguramente a problemas con el tipo de codificación en la que Waterfox está programado.



Il·lustració 37. Error 4 E Problemas con las tildes

Pero también podemos comentar que este botón sirve para abrir una ventana nueva, igual que un botón anterior. Entonces, ¿por qué está ahí? ¿Y por qué lo tradujeron de esa manera?



Il·lustració 38. Error 5 E El mismo problema con las tildes

Indagando en la carpeta de Waterfox que se genera al descargar el programa, hay algo curioso: el paquete de lengua se encuentra en formato.xpi. Según consultamos, es el formato que utiliza Firefox para incluir las distintas traducciones de sus productos. Tratamos de abrirlo para examinarlo, pero lo que vimos fue un texto codificado imposible de leer.

Más adelante descubrimos que ese formato era para archivos comprimidos. Lo copiamos en el escritorio y luego extrajimos lo que contenía mediante WinRAR.

El contenido constaba de varias carpetas y archivos, entre ellos los .dtd y los .properties que habíamos traducido, los cuales estaban en la siguiente extensión del paquete de lengua en GitHub: «langpack-ca@waterfox\browser\chrome\ca\locale\browser», la misma dentro de la cual Rubén encontró los archivos originales en la página de GitHub. De modo que probamos a sustituirlos por los archivos que habíamos traducido, tanto los que ya estaban allí como los que no se habían traducido al catalán. Desafortunadamente, no funcionó.

Tampoco sirvió extraer el contenido del .xpi en catalán y pegar los archivos traducidos dentro de las carpetas.

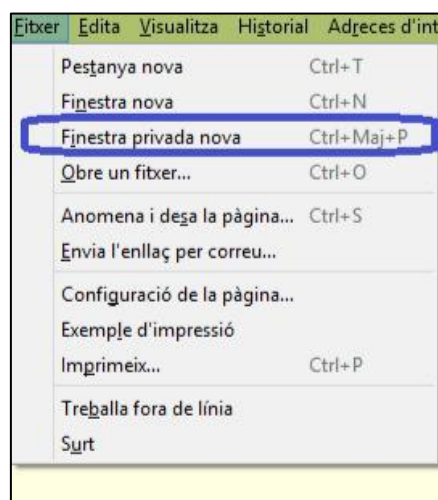
Además, tras observarlos vimos que este formato tiene un lenguaje que no admite ciertas gráficas por el tipo de código en el que se ha escrito, como las tildes y las diéresis, las cuales sustituye por la letra «d» o el número «10», tal como se muestra en las capturas anteriores y en la siguiente.

28	tabs.unmuteAudio.tooltip= No silenci <u>d's</u> la pestanya (%S)
29	tabs.muteAudio.background.tooltip= Silenci a la pestanya
30	tabs.unmuteAudio.background.tooltip= No silenci <u>d's</u> la pestanya

Il·lustració 39. Error 6 E Problemas con las diéresis

Y, por alguna razón, si se descomprime el .xpi, se modifica alguna palabra, y se vuelve a comprimir, al dejarlo en la misma carpeta y reinstalar el programa, este no reconoce el paquete de lengua y ya no está disponible en catalán o la lengua del paquete que se haya manipulado.

Tras todos estos tropiezos, nos metimos en la carpeta de Waterfox que se genera al descargar el programa y buscamos uno de los errores ya comentados, el de «e10s», lo modificamos y volvimos a comprimir todo el contenido para crear el .xpi. Estuvimos buscando cómo crearlo para que no hubiera errores y la solución que encontramos fue comprimir el contenido de la carpeta que se quisiera comprimir, no la carpeta que lo contenía. Sin embargo, el resultado no fue muy satisfactorio. El botón había desaparecido. Y fondo amarillo indicaba que hubo un error de análisis en el XML.



Il·lustració 40. Problema casi solucionado

Ambos lo volvimos a intentar más veces y lo que ocurría era que desaparecía la traducción y el programa pasaba a estar por completo en inglés. De arreglar estos problemas técnicos se encargaría un ingeniero o informático en el mundo profesional. Pero esto era un proyecto de clase y es posible que nos toque arreglar un caso similar algún día. Y, aparte, se trata de software libre y los que se dedican a la tradumática tienen que ocuparse de casi todo.

Le pedimos ayuda al profesor y vimos que habíamos cometido un error básico en la fase de testing. Al abrir el paquete de lengua no había que descomprimirlo. Bastaba con abrirlo mediante un programa de descompresión, como WinRAR o IZArc. Y, desde el programa, abrir los archivos con el texto traducible e ir modificándolos desde un procesador de textos como Notepad++. Una vez abiertos, podíamos copiar y pegar nuestras traducciones en los segmentos donde había texto sin traducir o con errores. Y de este modo no hubo problemas para que el programa arrancase con nuestras traducciones.

Si se hace bien, el procesador abrirá un campo de texto en el que se nos avisará de que estamos modificando un archivo del programa y que si queremos guardar los cambios, que es lo que queremos. Con lo cual, debemos de aceptar.

Es muy importante no tener abierto el navegador durante esta fase y utilizar el editor en modo administrador para evitar que surjan otros problemas. El resultado es el esperado. Por no mencionar que la traducción disponible es mejorable, como ya dijimos al principio.

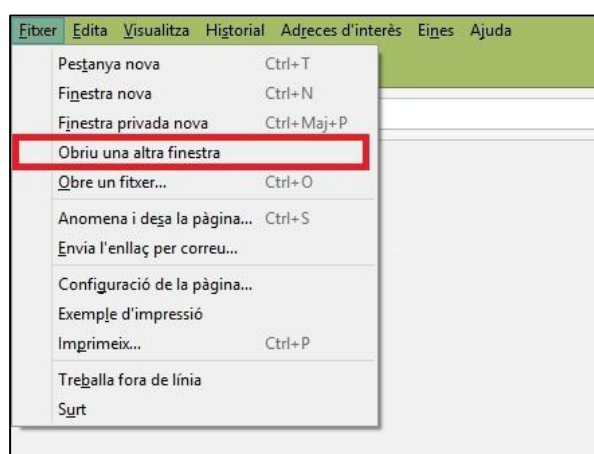


Ilustración 41. Problema solucionado



Ilustración 42. Ejemplo de lo que también seríamos capaces de traducir (catalá)



Ilustración 43. Ejemplo de lo que también seríamos capaces de traducir (chino)

7. Conclusiones

(Rubén)

Con este proyecto me he dado cuenta del trabajo que supone acarrear todas las tareas del proceso de traducción. Lo que me lleva a pensar que es una buena idea que en una empresa se distribuyan los roles. Y opino además que los traductores autónomos que hagan todo el trabajo por sí solos merecen mucho respeto.

Es curioso que, a lo largo de cuatro años de carrera, a duras penas se nos mencionen las herramientas de traducción. Trabajamos casi siempre sobre un procesador de textos, con algunos diccionarios y recursos online, pero no con una herramienta de traducción automática ni memorias de traducción. La realidad es que, sabiendo utilizar estos recursos, el trabajo se hace mucho más rápido. Gracias a memorias ya existentes o las que el traductor va alimentando a medida que trabaja, el proceso no solo se agiliza muchísimo, sino que se pueden establecer parámetros de calidad, como crear bases terminológicas para términos que se repetirán para evitar traducirlos de dos formas diferentes, de modo que el resultado tenga cohesión. Gracias a estas herramientas se puede trabajar muy rápido, de forma cómoda y obtener buenos resultados. Siento que pronto no concebiré traducir sin el apoyo de las memorias, las bases terminológicas y las herramientas TAO.

Mientras hacía este trabajo me di cuenta de la ventaja de las memorias de traducción. Gracias a ellas mi desempeño ha sido leve y sin embargo el resultado ha sido bueno. Ojalá potenciasen el uso de estas herramientas durante la carrera.

A decir verdad, la parte a la que he dedicado más tiempo y esfuerzo, dejando de lado los percances técnicos, ha sido la revisión. Si se tiene un buen nivel de las lenguas con las que se trabaja y se dispone de un buen ojo para detectar errores, se puede ser muy buen profesional. Y siempre, antes de cualquier entrega, se debe pasar el corrector y hacer el control de calidad.

Aún me queda todo un curso para seguir aprendiendo y optimizar mi forma de trabajar, de manera que pueda traducir todavía mejor y en menos tiempo, ya sean productos digitales o no. Como persona que se considera bastante autodidacta, incluso aprendería algo de programación si fuese necesario.

Ahora bien, prefiero trabajar con una herramienta que esté en local, para evitarme problemas relacionados con la conexión a internet. Ahora que poseo el programa OmegaT, no tardaré en usarlo para futuras traducciones. Aparte, Trados, MemoQ y Déjà Vu también me parecen buenas opciones.

El mayor problema que le veo a esta cuenta compartida de Memsources es su falta de privacidad. Y además el traductor automático de Tradumática no me fue muy útil, por desgracia, razón por la cual lo descarté.

Con respecto al tema de la localización, he aprendido que hay que observar muy bien el contenido de las carpetas que genera el programa al descargarse. Tal vez ahí se encuentren los paquetes de lengua y los archivos que quedan por traducir, o los archivos que contienen texto sin traducir o errores, y podamos extraer todos los archivos con texto para traducirlos, en vez de buscarlos en plataformas en internet que, por mucho que contengan material del desarrollador del producto que vayamos a traducir, no es lo mismo que poder manipular los propios archivos del producto que se generan durante la instalación.

De hecho, esto era lo que debí de haber hecho desde el principio, en lugar de traducir archivos que encontrara por internet. En fin, de los errores se aprende, y esto no se me va a olvidar. Aunque luego en el mundo profesional el gestor de proyectos me envíe los archivos, no está de más saber manejar otros recursos. Quizá incluso algún día trabaje como gestor.

También he aprendido que hay que ejecutar las herramientas que estemos utilizando en modo administrador; además de no utilizar el bloc de notas ni nada parecido, sino programas especializados, para realizar una fase de testing bien hecha. A decir verdad, es así como se pueden detectar los errores con mayor facilidad: observando el programa tal como lo hará nuestro cliente.

Mi siguiente paso será empezar a participar en esas páginas web como GitHub, que acepten mis propuestas de incluir traducciones con algunos errores solucionados y ver materializado mi trabajo, de modo que esté disponible para todo el mundo. Seguiré intentando que acepten mis propuestas después de realizar este proyecto.

(Xiao)

He aprendido mucho desde que empecé a estudiar el máster. Al principio ni siquiera sabía qué era una memoria de traducción, nunca había usado ninguna herramienta de traducción,

pero ahora ya tengo una idea básica de qué es la tradumática. He practicado con muchas herramientas de traducción, tales como OmegaT, Trados, MemoQ, Memsource, Déjà vu, Catalyst, Passolo, etc. También he aprendido a localizar aplicaciones y a cómo instalar un sistema operativo a través de un USB.

En el proceso de traducción hice tareas desde crear un nuevo proyecto, añadir los documentos por traducir, generar un informe de análisis básico y un análisis de concordancia, importar memorias de traducción ya existentes de Firefox, configurar TA, hacer pretraducción, traducir manualmente y corregir las coincidencias, hasta hacer el control de calidad, exportar el proyecto traducido, el glosario y la memoria de traducción. Incluso he aprendido a hacer un testing.

En este trabajo, la memoria de traducción de Firefox me ha servido de mucho, porque la mayoría de los segmentos por traducir tenían su correspondiente concordancia. Pero aún hay elementos que me parecen difíciles de traducir, especialmente los términos especializados del ámbito de la configuración del ordenador. Para mejorar mi traducción, estudié cómo introducir los motores de traducción automática en OmegaT, como Google Translator o Microsoft Translator. También he consultado mucho Glosbe y Linguee. Me van muy bien como referencias.

Mientras traducía, iba creando el glosario, que me recordaba cómo había traducido la misma palabra dentro de este proyecto. Por supuesto, el glosario también me servirá para traducir otro navegador en el futuro. Además, he practicado la función «buscar y reemplazar».

Y he aprendido cómo trabajar con las etiquitas no reconocibles en la traducción. También cometí errores, por ejemplo, no haber puesto la memoria de traducción de Firefox en la carpeta «auto» de «tm» para que aparecieran las concordancias automáticamente. Pero ya lo corregí y aprendí. Y en la fase de testing, aunque he tardado mucho tiempo en solucionar el tema de comprimir y guardar mis traducciones, cambiar al modo de administrador, etc., al final estaba muy feliz de ver el diálogo en chino de «about Waterfox» en el interfaz. Espero tener en el futuro más oportunidades de practicar la localización, quizás la pueda probar con un motor de traducción automática. Tengo confianza de que paso a paso, lograré localizar más tipos de software con muy buena calidad.

8. Bibliografia

https://en.wikipedia.org/wiki/List_of_free_and_open-source_software_packages
<https://hicksdesign.co.uk/journal/branding-Firefox>
<https://www.howtogeek.com/335712/update-why-you-shouldnt-use-waterfox-pale-moon-or-basilisk>
http://www.termcat.cat/ca/Comentaris_Terminologies/Consultes_Terminologiques/77
<http://www.waterfox.org>
<https://www.memsource.com>
<https://en.wikipedia.org/wiki/Waterfox>
<https://github.com/MrAlex94/Waterfox>
<https://www.softcatala.org/recursos/memories.html>
<https://www.softcatala.org/guia-estil-de-softcatala/aspectes-linguistics>
<http://aplicacions.llengua.gencat.cat/llc/AppJava/index.html>
<https://mdlc.iec.cat/>
https://developer.mozilla.org/en-US/docs/Mozilla/Localization/Localizing_without_a_specialized_tool
<https://stackoverflow.com/questions/19240653/how-to-create-xpi-file-with-7zip>
<http://omegat.org>
<https://zh.glosbe.com>
<https://www.linguee.es>
<https://omegat.sourceforge.io/manual-latest/es/chapter.dictionaries.html>
http://omegat.org/howtos/google_translate
<https://omegat.sourceforge.io/manual-latest/es/chapter.file.filters.html>